std::array Solutions

Copyright © James Raynard 2021

Built-in array

- What are the main advantages of built-in arrays over std::array?
  - Number of elements does not have to be known at compile time
  - Can be allocated on the heap if too big for the stack
  - Can be resized

Copyright © James Raynard 2021

std::array

- What are the main advantages of std::array over built-in arrays?
  - The number of elements is part of the type of std::array
  - Impossible to use an std::array with the wrong number of elements
  - std::array has a member function that returns the number of elements
  - std::array is never converted to a pointer
  - std::array is compatible with other STL containers
  - Supports range checking, using the at() member function
  - std::arrays can be assigned, if they have the same type
- In what situations would it be preferable to use a built-in array instead of either an std::array or std::vector?
  - If compatibility with old code is important

Copyright © James Raynard 2021

std::array usage

- Write a program which
  - Creates and initializes an std::array
  - Prints out an element of the array
  - Modifies an element of the array and prints out the result
  - Assigns one std::array object to another

Copyright © James Raynard 2021

Looping

- What forms of loops are available with std::array?
  - std::array supports iterators, so all forms of loop are supported
- Modify your program from the last exercise to print out the elements of the std::array. Use all the forms of loop which it supports

Copyright © James Raynard 2021

std::array interface

- Why the interface of std::array not exactly the same as for std::vector?
  - std::array is a fixed-size container, hence there are no member functions which add or remove elements
  - std::array does not container a memory buffer, so there are no member functions for managing the memory buffer

Copyright © James Raynard 2021

std::array as function argument

- Write a function which takes an std::array and prints out its elements
- Write an overload of this function which takes a built-in array
- Write a program to test your functions
- What happens if you pass an std::array with the wrong number of elements? Explain your results
  - The program does not compile
  - This is because the number of elements is part of the type of std::array
  - std::array objects with different numbers of elements have different types
  - To make this work, we would need to add another overload which takes an std::array which has the required number of elements

Copyright © James Raynard 2021